IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BOARD OF PATENT APPEALS AND INTERFERENCES

In Re Application of: Yen-Kuang Chen et al. Application No. 10/687,953 Filed: October 17, 2003 For: METHOD AND APPARATUS FOR EFFICIENT BI-LINEAR INTERPOLATION AND MOTION COMPENSATION	Examiner: David H. Malzahn Art Unit: 2123 **CERTIFICATE OF TRANSMISSION** I hereby certify that this correspondence is being transmitted to the United States Patent and Trademark Office, or facsimile transmitted to Fax No. (571)273-8300 on 3-17-2008 /Lawrence M. Mennemeier/ Date Lawrence M. Mennemeier

APPELLANT'S BRIEF UNDER 37 CFR § 41.37
IN SUPPORT OF APPELLANT'S APPEAL TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

Mail Stop Appeal Brief-Patents
Commissioner of Patents
PO Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Appellant hereby submits this Brief in support of an appeal from a final decision of the Examiner, in the above-referenced case. Appellant respectfully requests consideration of this appeal by the board of Patent Appeals and Interference for allowance of the above-referenced patent application.

TABLE OF CONTENTS

I. Real Party in Interest

The real party in interest in the present appeal is Intel Corporation of Santa Clara, California, the assignee of the present application.

II. Related Appeals and Interferences

There are no related appeals or interferences to appellant's knowledge that would have a bearing on any decision of the Board of Patent Appeals and Interferences.

III. Status of the Claims (independent claims shown in bold)

Claims **1**-29 are pending in the application.

Claims **1**-8, and 10-12 stand rejected under 35 USC § 101 as allegedly being directed to a program per se.

Claims **1**-8, 10-12 and **13**-18 stand rejected under 35 USC § 101 as allegedly being directed to a mathematical algorithm without a practical application.

Claims **19**-24 and **25**-29 are allowed.

Final rejection of claims **1**-8, 10-12 and **13**-18 is being appealed.

IV. Status of Amendments

An official amendment and response to a first Office Action mailed 1/8/2007 was submitted by appellant on 7/9/2007 and was entered. A Final Office Action was mailed on 8/17/2007. A Notice of Appeal was transmitted on 1/17/2007, and an appeal ensued. Another amendment is being submitted, under 37 CFR § 41.33 and concurrent with the present appeal brief.

Accordingly, the claims stand as of the concurrently submitted amendment of 3/17/2008, and are reproduced in clean form in the Claims Appendix.

V. Summary of Claimed Subject Matter

Appellant's disclosure describes methods and apparatus for performing bi-linear interpolation and motion compensation including multiply-add operations and byte shuffle operations on packed data in a processor. Bi-linear interpolation and motion compensation are techniques for decompression and display of digital images and video. In one embodiment, two or more lines of $2n+1$ content byte elements may be shuffled to generate a first and second packed data respectively including at least a first and a second $4n$ byte elements including $2n-1$ duplicated elements. A third packed data including sums of products is generated from the first packed data and packed byte coefficients by a multiply-add instruction. A fourth packed data including sums of products is generated from the second packed data and packed byte coefficients by another multiply-add instruction. Corresponding sums of products of the third and fourth packed data are then summed, and may be rounded and averaged.

Claim 13, for example, sets forth a machine-accessible medium[1] including data to perform a bi-linear interpolation or a motion compensation of a digital image or video[2] such that, when accessed by one or more machines, causes said one or more machines to: shuffle a first 2n+1 byte elements of a first line of data to generate a first packed data comprising at least a first 4n byte elements including 2n-1 duplicated elements of the first 2n+1 byte elements[3], shuffle a second 2n+1 byte elements of a second line of data to generate a second packed data comprising at least a second 4n byte elements including 2n-1 duplicated elements of the second 2n+1 byte elements[4], multiply-add the first packed

---

[1] "The present invention may be provided as a computer program product which may include a machine or computer-readable medium having stored thereon instructions which may be used to program a computer (or other electronic devices) to perform a process according to the present invention. The computer-readable medium may include, but is not limited to, floppy diskettes, optical disks, Compact Disc, Read-Only Memory (CD-ROMs), and magneto-optical disks, Read-Only Memory (ROMs), Random Access Memory (RAMs), Erasable Programmable Read-Only Memory (EPROMs), Electrically Erasable Programmable Read-Only Memory (EEPROMs), magnetic or optical cards, flash memory, or the like." **(US 2004/0139138 A1 p. 2, par. 30)**

[2] "Accordingly, the display of images, as well as playback of audio and video data, which is collectively referred to herein as content, have become increasingly popular applications for current computing devices. Bi-linear interpolation and motion compensation are popular techniques for decompression and display of images and video." **(US 2004/0139138 A1 p. 1, par. 4, lines 1-7)** "A method and apparatus for efficient bi-linear interpolation and motion compensation of content data are described." **(US 2004/0139138 A1 p. 2, par. 26, lines 1-3)** "**Figure 12** illustrates one exemplary embodiment of a bi-linear interpolation and motion compensation of content data." **(US 2004/0139138 A1 p. 14, par. 142, lines 1-3; Fig. 12)** "It will be appreciated that bi-linear interpolation may be broadly useful in many other image processing or video processing applications. Especially in applications where small integer data types are used and have sufficient representational range, the techniques detailed below may also be found useful." **(US 2004/0139138 A1 p. 14, par. 144, lines 9-14)**

[3] "In processing block 1321, a line with 2n+1 byte elements is shuffled to generate a packed data comprising 4n packed byte elements." **(US 2004/0139138 A1 p. 14, par. 147, lines 3-5; Fig. 13B, 1321)** "In processing block 1331, each of two lines, a first line with 2n+1 byte elements and a second line with 2n+1 byte elements are shuffled to generate packed data comprising a first 4n byte elements and a second 4n byte elements." **(US 2004/0139138 A1 p. 14, par. 148, lines 3-7; Fig. 13C, 1331)** "In processing block 1502 Result 1511 is shuffled according to according to packed data 1560 stored in SRC2 to generate Result 1612 comprising 2 lines of 4n shuffled elements each. Data of byte elements 1511 in the byte position indicated by said byte element of packed data 1560 is stored to Result 1612 in the byte position corresponding to said byte element of packed data 1560." **(US 2004/0139138 A1 p. 15, par. 158; Fig. 15A, 1502)**

[4] "In processing block 1323, another line with 2n+1 byte elements is shuffled to generate a packed data comprising a second 4n packed byte elements." **(US 2004/0139138 A1 p. 14, par. 147, lines 9-11; Fig. 13B, 1323)** "In processing block 1333, each of two lines, with 2n+1 byte elements are shuffled to generate more packed data comprising 4n byte elements for each of the last two line of 2n+1 bytes shuffled." **(US 2004/0139138 A1 p. 14, par. 148, lines 10-13; Fig. 13C, 1333)** "In processing block 1504 Result 1521 is shuffled according to according to packed data 1560 stored in SRC2 to generate Result 1622 comprising 2 lines of 4n shuffled elements each. Data of byte elements 1521 in the byte position indicated by said byte

data with at least a first two byte coefficients to generated a third packed data including sums of products[5]; multiply-add the second packed data with at least a second two byte coefficients to generated a fourth packed data including sums of products[6]; and add corresponding sums of products of the third and fourth packed data to generate a first packed result[7].

Claim 1 sets forth a method for performing a bi-linear interpolation or a motion compensation of a digital image or video[2], the method comprising: decoding a first shuffle instruction[8] and a first multiply-add instruction[9], each of an instruction format comprising

---

element of packed data 1560 is stored to Result 1622 in the byte position corresponding to said byte element of packed data 1560." **(US 2004/0139138 A1 p. 15, par. 160; Fig. 15B, 1504)**

[5] "In processing block 1322, a multiply-add operation is performed on the shuffled 4n packed byte data and at least two area coefficients to generated a packed data including 16-bit sums of products." **(US 2004/0139138 A1 p. 14, par. 147, lines 5-9; Fig. 13B, 1322)** "In processing block 1332, a multiply-add operation is performed on the shuffled packed byte data and at least two area coefficients to generated a packed data including 16-bit sums of products." **(US 2004/0139138 A1 p. 14, par. 148, lines 7-10; Fig. 13C, 1332)** "In processing block 11601, a multiply-add operation is performed on the shuffled packed byte data 1612 stored in SRC1 and at least two area coefficients 1640 stored in SRC2 to generated Result 1610 including 16-bit sums of products." **(US 2004/0139138 A1 p. 15, par. 161, lines 3-7; Fig. 16A, 1601)**

[6] "In processing block 1324, a multiply-add operation is performed on the second 4n packed byte data and at least two more area coefficients to generated another packed data including 16-bit sums of products." **(US 2004/0139138 A1 p. 14, par. 147, lines 11-15; Fig. 13B, 1324)** "In processing block 1334, a multiply-add operation is performed on the second shuffled packed byte data and at least two more area coefficients to generated another packed data including 16-bit sums of products." **(US 2004/0139138 A1 p. 14, par. 148, lines 13-17; Fig. 13C, 1334)** "In processing block 1602, a multiply-add operation is performed on the second shuffled packed byte data 1622 stored in SRC1 and at least two more area coefficients 1650 stored in SRC2 to generated Result 1620 including 16-bit sums of products." **(US 2004/0139138 A1 p. 15, par. 161, lines 8-12; Fig. 16A, 1602)**

[7] "Processing proceeds to processing block 1325 where a packed result including 2n sums is generated by adding corresponding elements of the last two packed 16-bit sums of products generated." **(US 2004/0139138 A1 p. 14, par. 147, lines 15-18; Fig. 13B, 1325)** "Processing proceeds to processing block 1335 where a packed result including 4n sums is generated by adding corresponding elements of the last two packed 16-bit sums of products generated." **(US 2004/0139138 A1 p. 14, par. 148, line 17 through p. 15, line 2; Fig. 13C, 1335)** "Processing proceeds to processing block 1603 where a Result 1611 including 4n sums is generated by adding corresponding elements of the packed 16-bit sums of products of Result 1610 and Result 1620." **(US 2004/0139138 A1 p. 15, par. 161, lines 12-15; Fig. 16A, 1603)**

[8] "In processing block 1001, decoder 165 decodes the control signal received by processor 109. Thus, decoder 165 decodes the operation code for a shuffle instruction." **(US 2004/0139138 A1 p. 12, par. 131, lines 3-6; Fig. 10A, 1001)** "In processing block 1001 through 1003 processing is substantially similar to that of process 1011 if the instruction is a shuffle instruction for processing 64-bits of packed byte data." **(US 2004/0139138 A1 p. 13, par. 134, lines 3-6; Fig. 10B, 1001)** "In processing block 1001, decoder 165 decodes the control signal received by processor 109. Thus, decoder 165 decodes the operation code for a shuffle instruction." **(US 2004/0139138 A1 p. 13, par. 137, lines 3-6; Fig. 10C, 1001)**

[9] "In processing block 701, decoder 165 decodes the control signal received by processor 109. Thus, decoder 165 decodes the operation code for a multiply-add instruction. It will be appreciated that while

a first operand field and a second operand field[10]; responsive at least in part to said first shuffle instruction, generating a first packed data having a first plurality of byte data elements including an a1 byte data element, and at least two copies of each of a2, a3, and a4 byte data elements[3,4]; and responsive to said first multiply-add instruction, wherein the first operand field of said first multiply-add instruction specifies said first packed data and the second operand field specifies a second packed data having a second plurality of byte data elements including at least two copies of each of b1 and b2 byte data elements, performing an operation (a1 x b1) + (a2 x b2) to generate a first 16-bit data element of a

---

examples are given of multiply-add, one of skill in the art could modify the teachings of this disclosure to also perform multiply-subtract without departing from the broader spirit of the invention as set forth in the accompanying claims." **(US 2004/0139138 A1 p. 9, par. 96; Fig. 7A, 701)** "Processing blocks 701 through 703 are essentially the same as in process block 711, with the exception that in processing block 703, the instruction is a multiply-add instruction for performing byte multiplications on 128-bit packed data, and so flow passes to processing block 719." **(US 2004/0139138 A1 p. 10, par. 103, lines 3-8; Fig. 7B, 701)**

[10] "Up to three operand locations per instruction may be identified, including up to two source operand identifiers SRC1 602 and SRC2 603 and one destination operand identifier DEST 605." **(US 2004/0139138 A1 p. 7, par. 80, lines 11-14; Fig. 6A, 602, 603)** "Up to two operand locations per instruction may be identified, including up to two source operand identifiers SRC1 602 and SRC2 603. For one embodiment of the multiply-add/subtract instruction and the shuffle instruction, destination operand identifier DEST 605 is the same as source operand identifier SRC1 602. For an alternative embodiment, destination operand identifier DEST 605 is the same as source operand identifier SRC2 603. Therefore, for embodiments of the multiply-add/subtract operations and/or the shuffle operations, one of the source operands identified by source operand identifiers SRC1 602 and SRC2 603 is overwritten by the results of the multiply-add/subtract operations or the shuffle operations." **(US 2004/0139138 A1 p. 7, par. 82, lines 10-22; Fig. 6B, 602, 603)** "The type of operation, may be encoded by one or more of fields 632 and 634 and up to two operand locations per instruction may be identified, including up to two source operand identifiers SRC1 602 and SRC2 603. For example, in one embodiment of the multiply-add instruction, field 632 may be set to a hexadecimal value of 0F38 and field 634 may be set to a hexadecimal value of 04 to indicate that data associated with source operand identifier SRC1 602 is to be treated as unsigned packed bytes, data associated with source operand identifier SRC2 603 is to be treated as signed packed bytes and result data associated with destination operand identifier DEST 605 is to be treated as signed packed words. In one embodiment of the shuffle instruction, field 632 may be set to a hexadecimal value of 0F38 and field 634 may be set to a hexadecimal value of 00 to indicate that byte data is associated with source operand identifier SRC1 602 is to be reordered according to byte fields associated with source operand identifier SRC2 603 and stored as packed byte data associated with destination operand identifier DEST 605." **(US 2004/0139138 A1 p. 8, par. 83, lines 3-23; Fig. 6C, 602, 603)** "Up to two operand locations per instruction may be identified by source operand identifiers SRC1 602 and SRC2 603 and by prefix byte 640. For one embodiment of the multiply-add/subtract instruction and/or the shuffle instruction, prefix byte 640 may be used to identify 128-bit source and destination operands. For example, in one embodiment of the multiply-add instruction and/or the shuffle instruction, prefix byte 640 may be set to a hexadecimal value of 66, to indicate that 128 bits of data from one of the extension registers 210 are associated with source operand identifiers SRC1 602 and SRC2 603 and 128 bits of result data from one of the extension registers 210 are associated with destination operand identifier DEST 605." **(US 2004/0139138 A1 p. 8, par. 88, lines 7-19; Fig. 6D, 602, 603)**

third packed data, performing an operation (a2 x b1) + (a3 x b2) to generate a second 16-bit data element of the third packed data, and performing an operation (a3 x b1) + (a4 x b2) to generate a third 16-bit data element of the third packed data[5,6].

VI. Grounds of Rejection to be Reviewed on Appeal

A. Claims 1-8, and 10-12 stand rejected under 35 USC § 101 as allegedly being directed to a program per se.

B. Claims 1-8, 10-12 and 13-18 stand rejected under 35 USC § 101 as allegedly being directed to a mathematical algorithm without a practical application.

VII. Argument

A. CLAIMS 1-8 AND 10-12 ARE DIRECTED TO STATUTORY SUBJECT MATTER

Claims 1-8 and 10-12 stand rejected under 35 USC § 101 as allegedly being directed to a program per se, because no computer-readable medium is recited for recording functionally descriptive material.

With regard to process claims 1-8, and 10-12, appellant respectfully submits that the examiner is in error for concluding that they are directed to a program per se. Claim 1 sets forth:

> 1. (Previously Presented) A method for performing a bi-linear interpolation or a motion compensation of a digital image or video, the method comprising:
>
> decoding a first shuffle instruction and a first multiply-add instruction, each of an instruction format comprising a first operand field and a second operand field;
>
> responsive at least in part to said first shuffle instruction, generating a first packed data having a first plurality of byte data elements including an a1 byte data element, and at least two copies of each of a2, a3, and a4 byte data elements; and
>
> responsive to said first multiply-add instruction, wherein the first operand field of said first multiply-add instruction specifies said first packed data and the second operand field specifies a second packed data having a second plurality of byte data elements including at least two copies of each of b1 and b2 byte data elements, performing an operation (a1 x b1) + (a2 x b2) to generate a first 16-bit data element of a third packed data, performing an operation (a2 x b1) + (a3 x b2) to generate a second 16-bit data element of the third packed data, and performing an operation (a3 x b1) + (a4 x b2) to generate a third 16-bit data element of the third packed data.

An analysis of the instant claims must be performed in order to make a determination of whether the subject matter is statutory. Such analysis should correlate each claim element with corresponding structures, materials or acts set forth in the specification.

For example, page 2, par. 29, of the specification (emphasis added) discloses that:

> In one embodiment, methods of the present invention are embodied in machine-executable instructions. The instructions can be used to cause a general-purpose or special-purpose processor that is programmed with the instructions to perform the steps of the method. Alternatively, the steps of the method might be performed by specific hardware components that contain hardwired logic for performing the steps, or by any combination of programmed computer components and custom hardware components.

Therefore, appellant respectfully submits that the examiner is in error for miscategorizing the instant claim as a mere program listing rather than as a statutory process or improvement thereof.

Appellant respectfully submits that, following the examiner's reasoning, any process that could be automated through programmed machines would be non-statutory as a program per se. Indeed it may very well be the case that any process which can be listed as steps in a claim can in fact be automated by programmed machines. Therefore, according to the examiner's reasoning, any process which can be listed as steps in a claim is not patentable.

To this point, the Federal Circuit explained in *AT & T Corp. v. Excel Communications, Inc.* 172 F.3d 1352, 1356, 50 USPQ2D 1447, 1450 (Fed. Cir. 1999) that (emphasis added):

> This court recently pointed out that any step-by-step process, be it electronic, chemical, or mechanical, involves an "algorithm" in the broad sense of the term. *See State Street Bank & Trust Co. v. Signature Fin. Group, Inc.*, 149 F.3d 1368, 1374-75, 47 USPQ2d 1596, 1602 (Fed. Cir. 1998).

Yet 35 U.S.C. §101 clearly sets forth that (emphasis added), "Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title."

Thus, contrary to the examiner's assertion, the intent of such broad coverage for "any new and useful process" does not prohibit some particular new and useful processes simply because they can be performed by, or can make use of a computer program.

Therefore, appellant respectfully submits that the Examiner is in error for incorrectly concluding that the processes of claims **1-8**, and 10-12 are directed to programs per se (i.e. programs—of, in or by themselves, their functionally descriptive material standing alone without reference to additional means which would allow said functionality to be realized). Indeed, the MPEP §2106.01 further states that (emphasis added):

> Computer programs are often recited as part of a claim. USPTO personnel should determine whether the computer program is being claimed as part of an otherwise statutory manufacture or machine. In such a case, the claim remains statutory irrespective of the fact that a computer program is included in the claim. The same result occurs when a computer program is used in a computerized process where the computer executes the instructions set forth in the computer program. Only when the claimed invention taken as a whole is directed to a mere program listing, i.e., to only its description or expression, is it descriptive material *per se* and hence nonstatutory.

Since the instant claims are not directed to a mere program listing but rather to otherwise statutory processes, appellant respectfully submits that the claimed subject matter is statutory.

The final Office Action (p. 2, par. 3) further asserts that the instant claims are non-statutory for being directed to a data transformation, stating that, "the result is not a tangible result because it is not a real world result."

This notion has sometimes been phrased in terms of requiring a transformation or reduction of 'subject matter.' In Schrader, the phrase 'subject matter' was determined not to be limited to tangible articles or objects, but includes intangible subject matter, such as data or signals (e.g. in digital images and video), representative of or constituting physical activity or objects. *In re Schrader*, 22 F.3d at 295, 30 USPQ2D 1455 at 1459 (Fed. Cir. 1994).

Therefore, since the data of digital images and video being transformed is representative of or constituting physical activity or objects, appellant respectfully submits that Claims 1-8 and 10-12 are directed to statutory subject matter.

B. CLAIMS 1-8, 10-12 AND 13-18 ARE DIRECTED TO STATUTORY SUBJECT MATTER

Claims 1-8, 10-12 and 13-18 stand rejected under 35 USC § 101 as allegedly being directed to a mathematical algorithm without a practical application. The final Office Action states (p. 2, par. 3) that the method or process set forth by Claims 1-8, 10-12 and 13-18 do not produce a tangible result because it is not a real-world result, i.e. merely a data transformation. Appellant respectfully notes that the Examiner gives no controlling president or law to support this assertion.

On the other hand, as appellant has pointed out above, the notion of requiring a transformation or reduction of 'subject matter' to a different state or thing was addressed by the Federal Circuit in Schrader. In Schrader, the phrase 'subject matter' was determined not to be limited to tangible articles or objects, but includes intangible subject matter, such as data or signals, representative of or constituting physical activity or objects. *Schrader*, 22 F.3d at 295, 30 USPQ2D (BNA) at 1459.

The Federal Circuit clarified the meaning of the term 'subject matter' in Schrader (*Id.* at 295 n.12) and corrected a misconception that would require changes to a physical object, stating that (emphasis added):

> Professor Robinson cites to *Cochrane* for the above definition but inexplicably speaks in terms of changes to a physical "object" while *Cochrane* speaks in terms of changes to "subject matter." The distinction is significant. In the *Telephone Cases*, 126 U.S. 1 (1887), the Court upheld the validity of a claim directed to a method for transmitting speech by impressing acoustic vibrations representative of speech onto electrical signals. If there was a requirement that a physical *object* be transformed or reduced, the claim would not have been patentable. The point was recognized by our predecessor court in *In re Prater*, 415 F.2d 1393, 162 USPQ 541, 549 (CCPA 1969): "[The *Cochrane* passage] has sometimes been misconstrued as a 'rule' or 'definition' requiring that all processes, to be patentable, must operate physically upon substances. Such a result misapprehends the nature of the passage... ." *Id.* at 1403, 162 USPQ at 549, *modifying on rehearing*, 415 F.2d at 1387-88, 159 USPQ 583, 592 (CCPA 1968); *see also In re Musgrave*, 431 F.2d 882, 892, 167 USPQ 280, 289 (CCPA 1970). Thus, it is apparent that changes to *intangible* subject matter representative of or constituting physical activity or objects are included in the definition. *See Tilghman v. Proctor*, 102 U.S. 707, 728 (1881); *Corning v. Burden*, 56 U.S. (15 How.) 252 (1854).

Therefore, appellant respectfully submits that the Examiner is in error for asserting that a transformation of an article to a different sate or thing must be limited to tangible articles or objects and not merely a data transformation. If the data or signals being transformed are subject matter representative of or constituting physical activity or objects then a claimed method, process or article to facilitate their transformation to a different sate or thing is statutory subject matter.

The final Office Action also states (p. 2, par. 3) that the method or process set forth by Claims 1-8, 10-12 and 13-18 are directed not to a practical application, but to a mathematical algorithm to produce a mathematical result.

The Federal Circuit makes it clear that the ordinary and customary meaning of a claim term is the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention. "The person of ordinary skill in the art is deemed to read the claim term not only in the context of the particular claim in which the disputed term appears, but in the context of the entire patent, including the specification." *Phillips v. AWH Corp.*, 415 F.3d at 1313.

Appellant respectfully submits that the instant language, when correlated with the corresponding structures and processes set forth in the specification (e.g. Figs. 12, 13a-c, 14a-b, 15a-b and 16a-b; pars. 4-5, 26-29, and 142-161) makes it apparent to one of skill in the art that the claimed invention has a practical application in the technical arts, i.e. efficient bi-linear interpolation and motion compensation of content data for decompression of images and video.

Such practical applications in the technical arts include but are not limited to fractional-pixel chrominance motion compensation in accordance with the H.264

standard proposed by the Joint Video Team of the International Telecommunication Union (ITU) Video Coding Experts Group (VCEG) and the International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) Motion Picture Experts Group (MPEG) for decompression and display of images and video, which are representative of or constitute physical activity or objects. Therefore a claimed method, process or article to facilitate their transformation to a different sate or thing is statutory subject matter.

In addition, appellant respectfully submits, that the present application clearly asserts such a practical application in the technical arts.

For example, paragraphs 4-5 of the specification (emphasis added) assert that:

> Accordingly, the display of images, as well as playback of audio and video data, which is collectively referred to herein as content, have become increasingly popular applications for current computing devices. Bi-linear interpolation and motion compensation are popular techniques for decompression and display of images and video. Quarter-pixel and eighth-pixel motion compensation of luminance content in prior decompression techniques have made use of Finite Impulse Response (FIR) filters for interpolation. However for fractional-pixel chrominance motion compensation, bi-linear interpolation may be used instead.
>
> Recently bi-linear interpolations have been proposed by the Joint Video Team of the International Telecommunication Union (ITU) Video Coding Experts Group (VCEG) and the International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) Motion Picture Experts Group (MPEG) for fractional-pixel chrominance motion compensation in accordance with the H.264 standard (see Final Committee Draft of ISO/IEC 14496-10 Advanced Video Coding). Although the number of chrominance components is typically only half of the number of luminance components, the processing required for motion compensation of both types of components may be substantially equal.

Paragraphs 26 of the specification (emphasis added) assert that:

> A method and apparatus for efficient bi-linear interpolation and motion compensation of content data are described. In one embodiment, two or more lines of 2n+1 content byte elements may be shuffled to generate a first and second packed data respectively including at least a first and a second 4n byte elements including 2n-1 duplicated elements. A third packed data including sums of products is generated from the first packed data and packed byte coefficients by a multiply-add instruction. A fourth packed data including sums of products is generated from the second packed data and elements and packed byte coefficients by another multiply-add instruction. Corresponding sums of products of the third and fourth packed data are then summed, and may be rounded and averaged.

Paragraphs 142 of the specification (emphasis added) assert that:

> **Figure 12** illustrates one exemplary embodiment of a bi-linear interpolation and motion compensation of content data. Figure 12 illustrates a first line of 2n+1 bytes (a, b, c, d and e) and a second line of 2n+1 bytes (f, g, h, i, and j). Bi-linear interpolation and motion compensation may be performed, for example at quarter-pixel or eighth-pixel luminance resolution, on sub-sampled chrominance values to obtain 2n interpolated values (A, B, C and D).

Paragraphs 144 of the specification (emphasis added) assert that:

> Bi-linear interpolations have been proposed by the Joint Video Team of the International Telecommunication Union (ITU) Video Coding Experts Group (VCEG) and the International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) Motion Picture Experts Group (MPEG) for fractional-pixel chrominance motion compensation in accordance with the H.264 standard (see Final Committee Draft of ISO/IEC 14496-10 Advanced Video Coding). It will be appreciated that bi-linear interpolation may be broadly useful in many other image processing or video processing applications. Especially in applications where small integer data types are used and have sufficient representational range, the techniques detailed below may also be found useful.

Thus the specification makes it readily apparent to one of skill in the art that the claimed invention has a practical application in the technical arts.

The Supreme Court held that the focus in any statutory subject matter analysis be on the claim as a whole, stating "When a claim containing a mathematical formula implements or applies that formula in a structure or process which, when considered as a whole, is performing a function which the patent laws were designed to protect (e.g., transforming or reducing an article to a different state or thing, then the claim satisfies the requirements of § 101." *In re Alappat*, 33 F.3d 1526, 1543 (Fed. Cir. 1994) (quoting *Diehr*, 450 U.S. at 192, 209 USPQ at 10).

Therefore, appellant respectfully submits that since Claims **1**-8, 10-12 and **13**-18 implement or apply a mathematical formula in a process that transforms or reduces subject matter, which is representative of or constitute physical activity or objects, to a different state or thing, then the claims satisfy the requirements of 35 USC § 101. Accordingly, appellant respectfully submits that Claims **1**-8, 10-12 and **13**-18 are directed to statutory subject matter.

Conclusion

Appellant submits that all claims now pending are in condition for allowance. Such action is earnestly solicited at the earliest possible date. If there is a deficiency in fees, please charge our Deposit Acct. No. 50-0221.

Respectfully submitted,

Date: March 17, 2008

/Lawrence M. Mennemeier/
Lawrence M. Mennemeier
Reg. No. 51,003

INTEL CORPORATION
c/o INTELLEVATE LLP
P.O. Box 52050
Minneapolis, MN 55402
(408) 765-2194

VIII. Claims Appendix: Claims Allowed and Involved in Appeal (Clean Copy)

1. (Previously Presented) A method for performing a bi-linear interpolation or a motion compensation of a digital image or video, the method comprising:

decoding a first shuffle instruction and a first multiply-add instruction, each of an instruction format comprising a first operand field and a second operand field;

responsive at least in part to said first shuffle instruction, generating a first packed data having a first plurality of byte data elements including an a_1 byte data element, and at least two copies of each of a_2, a_3, and a_4 byte data elements; and

responsive to said first multiply-add instruction, wherein the first operand field of said first multiply-add instruction specifies said first packed data and the second operand field specifies a second packed data having a second plurality of byte data elements including at least two copies of each of b_1 and b_2 byte data elements, performing an operation $(a_1 \times b_1) + (a_2 \times b_2)$ to generate a first 16-bit data element of a third packed data, performing an operation $(a_2 \times b_1) + (a_3 \times b_2)$ to generate a second 16-bit data element of the third packed data, and performing an operation $(a_3 \times b_1) + (a_4 \times b_2)$ to generate a third 16-bit data element of the third packed data.

2. (Original) The method of Claim 1, further comprising:

decoding a second shuffle instruction and a second multiply-add instruction, each of an instruction format comprising a first operand field and a second operand field;

responsive at least in part to said second shuffle instruction, generating a fourth

packed data having a fourth plurality of byte data elements including a c1 byte data element, and at least two copies of each of c2, c3, and c4 byte data elements; and

responsive said second multiply-add instruction, wherein the first operand field of said second multiply-add instruction specifies said fourth packed data and the second operand field of said second multiply-add instruction specifies a fifth packed data having a fifth plurality of byte data elements including at least two copies of each of d1 and d2 byte data elements, performing an operation (c1 x d1) + (c2 x d2) to generate a first 16-bit data element of a sixth packed data, performing an operation (c2 x d1) + (c3 x d2) to generate a second 16-bit data element of the sixth packed data, and performing an operation (c3 x d1) + (c4 x d2) to generate a third 16-bit data element of the sixth packed data.

3. (Original) The method of Claim 2, further comprising:

decoding a packed add instruction of an instruction format comprising a first operand field and a second operand field;

responsive to said packed add instruction, wherein the first operand field of said packed add instruction specifies said third packed data and the second operand field of said packed add instruction specifies the sixth packed data, adding the first 16-bit data elements of the third and sixth packed data to generate a first 16-bit data element of a seventh packed data, adding the second 16-bit data elements of the third and sixth packed data to generate a second 16-bit data element of the seventh packed data, and adding the third 16-bit data elements of the third and sixth packed data to generate a third 16-bit data element of the seventh packed data.

4. (Original) The method of claim 1, said first plurality of byte data elements including at least 16 byte data elements and said second plurality of data elements including at least 16 byte data elements.

5. (Previously Presented) The method of claim 1, said first plurality of byte data elements including 8 byte data elements and said second plurality of data elements including 8 byte data elements

6. (Original) The method of claim 1 wherein said first plurality of data elements are treated as unsigned bytes.

7. (Original) The method of claim 6 wherein said second plurality of data elements are treated as signed bytes.

8. (Original) The method of claim 7 wherein each of said first, second and third 16-bit data elements are generated using signed saturation.

9. (Original) An apparatus to perform the method of Claim 8 comprising:

an execution unit including one or more execution circuits to execute operations on packed data elements;

at least one state machine; and

a machine-accessible medium including data that, when accessed by said at least one

state machine, causes said at least one state machine to enable the one or more execution circuits to perform the method of Claim 9.

10. (Original) The method of claim 1 wherein said first operand field comprises bits five through three of the instruction format.

11. (Original) The method of claim 10 wherein said second operand field comprises bits two through zero of the instruction format.

12. (Original) The method of claim 11 wherein said first plurality of byte data elements is overwritten by said third packed data responsive to the first multiply-add instruction.

13. (Original) A machine-accessible medium including data to perform a bi-linear interpolation or a motion compensation of a digital image or video such that, when accessed by one or more machines, causes said one or more machines to:

shuffle a first 2n+1 byte elements of a first line of data to generate a first packed data comprising at least a first 4n byte elements including 2n-1 duplicated elements of the first 2n+1 byte elements;

shuffle a second 2n+1 byte elements of a second line of data to generate a second packed data comprising at least a second 4n byte elements including 2n-1 duplicated elements of the second 2n+1 byte elements;

multiply-add the first packed data with at least a first two byte coefficients to generated a third packed data including sums of products;

multiply-add the second packed data with at least a second two byte coefficients to generated a fourth packed data including sums of products; and

add corresponding sums of products of the third and fourth packed data to generate a first packed result.

14. (Original) The machine-accessible medium of claim 13 including data that, when accessed by said one or more machines, causes said one or more machines to treat elements of the first packed data and of the second packed data as unsigned bytes in generating the sums of products of the third packed data and of the fourth packed data respectively.

15. (Original) The machine-accessible medium of claim 14 including data that, when

accessed by said one or more machines, causes said one or more machines to treat elements of said at least the first two byte coefficients and of said at least the second two byte coefficients as signed bytes in generating the sums of products of the third packed data and of the fourth packed data respectively.

16. (Original) The machine-accessible medium of claim 14 including data that, when accessed by said one or more machines, causes said one or more machines to overwrite the first packed data with the third packed data and to overwrite the second packed data with the fourth packed data.

17. (Original) The machine-accessible medium of Claim 13 including data that, when accessed by said one or more machines, further causes said one or more machines to:

shuffle the second 2n+1 byte elements of the second line of data to generate the first packed data comprising at least the second 4n byte elements including 2n-1 duplicated elements of the second 2n+1 byte elements; and

shuffle a third 2n+1 byte elements of a third line of data to generate the second packed data comprising at least a third 4n byte elements including 2n-1 duplicated elements of the third 2n+1 byte elements.

18. (Original) The machine-accessible medium of Claim 17 wherein n is equal to 2 and including data that, when accessed by said one or more machines, further causes said one or more machines to:

generate a second packed result comprising at least 4n rounded averages, each

corresponding to an element of said first packed result.

19. (Original) An apparatus comprising:

a decoder to decode a plurality of instructions including a shuffle instruction and a multiply-add instruction;

an execution unit including a first execution circuit, enabled by the decoded shuffle instruction, to shuffle a first 2n+1 byte elements of a first line of data to generate a first packed data comprising at least a first 4n byte elements including 2n-1 duplicated elements of the first 2n+1 byte elements, said execution unit further including a second execution circuit, enabled by the decoded multiply-add instruction, to multiply each of a first pair of byte data elements of the first packed data with a first pair of respective byte coefficient elements and to generate a first 16-bit result representing a first sum of products of the first pair of multiplications, and to multiply each of a second pair of byte data elements of the first packed data with a second pair of respective byte coefficient elements and to generate a second 16-bit result representing a second sum of products of the second pair of multiplications;

a first register to store the first packed data in response to the shuffle instruction; and

a second register to store a third packed data comprising at least said first and second 16-bit results in response to the multiply-add instruction.

20. (Original) The apparatus of claim 19 wherein n is at least two.

21. (Original) The apparatus of claim 20 wherein said first packed data contains at least

sixteen byte data elements.

22. (Original) The apparatus of claim 19 wherein the first packed data comprises unsigned byte data elements.

23. (Original) The apparatus of claim 22 wherein said first and second pairs of respective byte coefficient elements comprise signed byte data elements.

24. (Original) The apparatus of claim 19 wherein the first and second 16-bit results are generated using signed saturation.

25. (Original) A computing system comprising:

an addressable memory to store data;

a processor including:

a first storage area to store byte data elements of a content data;

a second storage area to store byte coefficient elements;

a decoder to decode a plurality of instructions including a first and a second shuffle instruction and a first and a second multiply-add instruction;

a first execution circuit, enabled by the decoded first shuffle instruction, to shuffle a first 2n+1 byte elements of the content data to generate a first packed data comprising at least a first 4n byte elements including 2n-1 duplicated elements of the first 2n+1 byte elements, and enabled by the decoded second shuffle instruction, to shuffle a second 2n+1 byte elements of the content data to generate a second packed data comprising at

least a second 4n byte elements including 2n-1 duplicated elements of the second 2n+1 byte elements;

a second execution circuit, enabled by the decoded first multiply-add instruction, to multiply each of a first and second pair of byte data elements of the first packed data with respective byte coefficient elements of a third packed data and to generate a first and a second 16-bit result respectively representing a first and a second sum of products of the first and the second pair of multiplications, and enabled by the decoded second multiply-add instruction, to multiply each of a third and fourth pair of byte data elements of the second packed data with respective byte coefficient elements of a fourth packed data and to generate a third and a fourth 16-bit result respectively representing a third and a fourth sum of products of the third and the fourth pair of multiplications;

a third storage area to store packed 16-bit data elements including the first and the second 16-bit results responsive to the first multiply-add instruction; and

a fourth storage area to store packed 16-bit data elements including the third and the fourth 16-bit result responsive to the second multiply-add instruction; and

a storage device to store said plurality of instructions.

26. (Original) The computing system of claim 25 wherein each of the first, second, third and fourth packed data comprise 16 byte data elements.

27. (Original) The computing system of claim 25 wherein each of the first, second, third and fourth packed data comprise 8 byte data elements.

28. (Original) The computing system of claim 25 wherein each of said first, second, third and fourth 16-bit results are generated by multiplying unsigned bytes elements of the content data by signed coefficient elements and by adding pairs of products with signed saturation.

29. (Original) The computing system of claim 25 wherein said plurality of instructions further includes an unpack instruction to store the byte data elements of the content data in the first storage area and a packed add instruction to add corresponding 16-bit data elements of the third and fourth storage areas.

IX. Evidence Appendix: With Copies of Evidence Relied Upon by Appellant

Appellant relies upon no additional evidence in this appeal.

X. Related Proceedings Appendix: Copies of Decisions Rendered by a Court or the Board in any Prior and Pending Appeals, Interferences or Judicial Proceedings

There are no related appeals or interferences to appellant's knowledge that would have a bearing on any decision of the Board of Patent Appeals and Interferences.